

FIG. 3

800

```

public BankKey ejbFindByPrimaryKey(
    java.lang.Integer arg0)
    throws javax.ejb.FinderException{
    return (BankKey) instanceExtension.FindByPrimaryKey(arg0);
}

```

FIG. 8

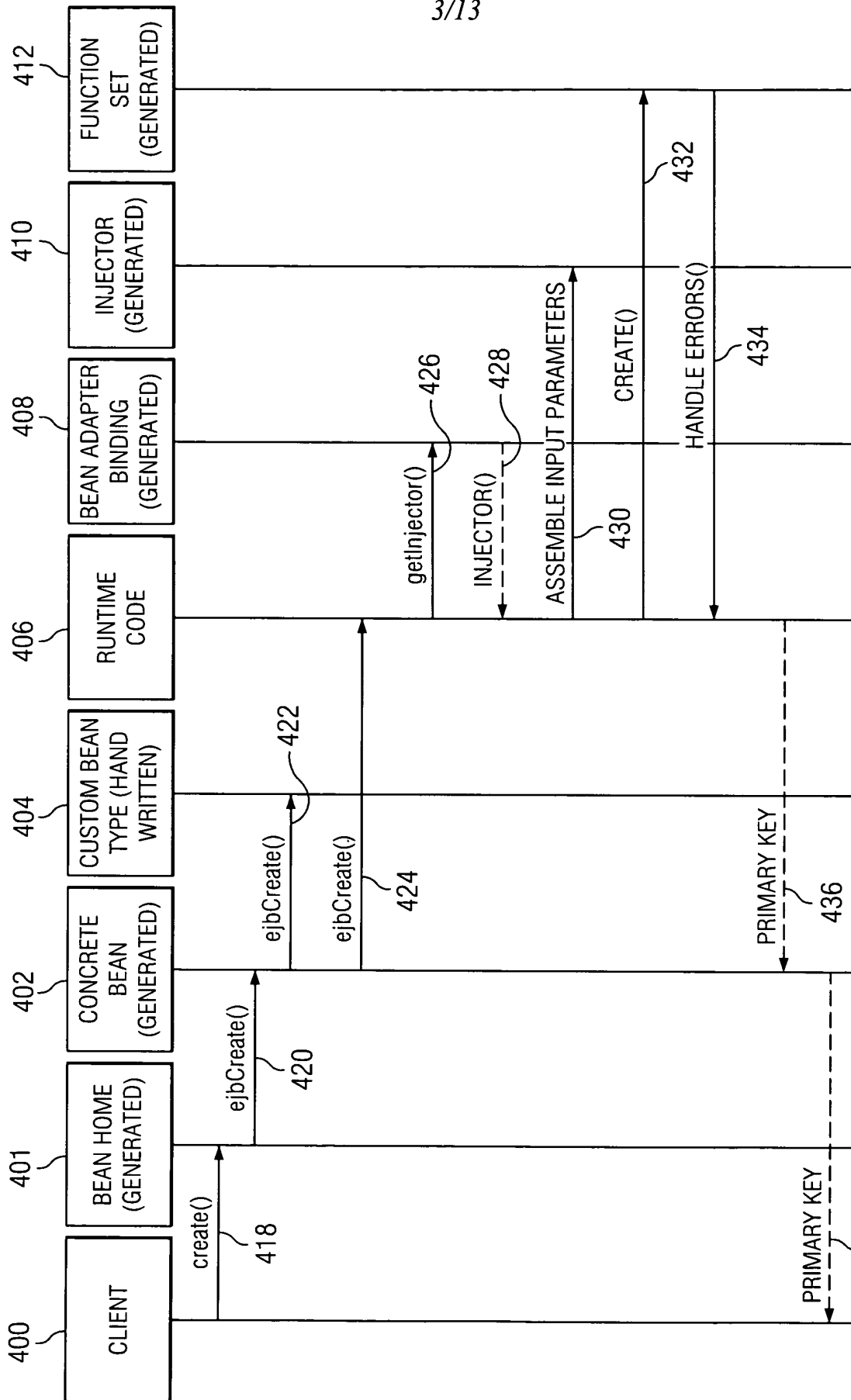


FIG. 4

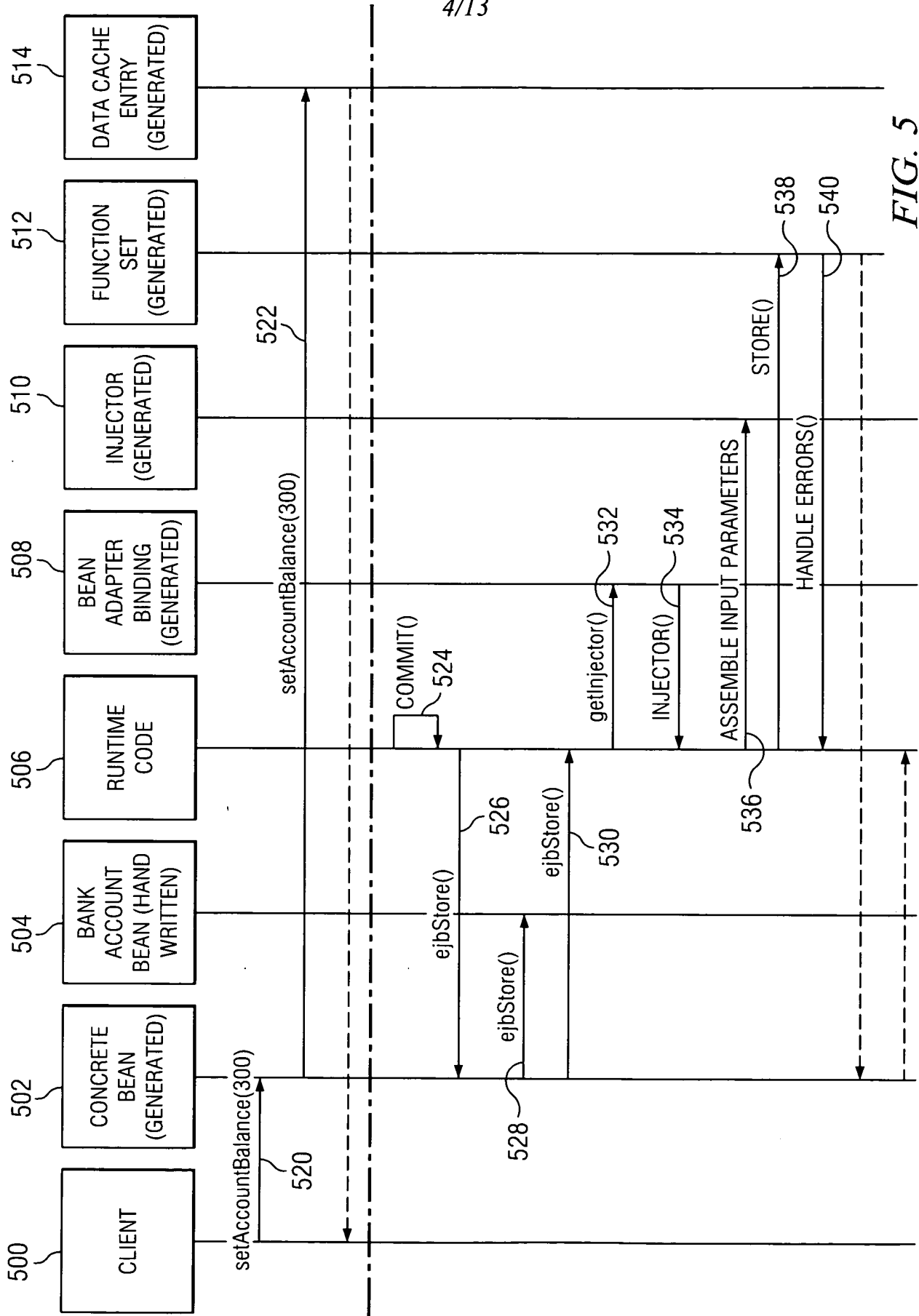


FIG. 5

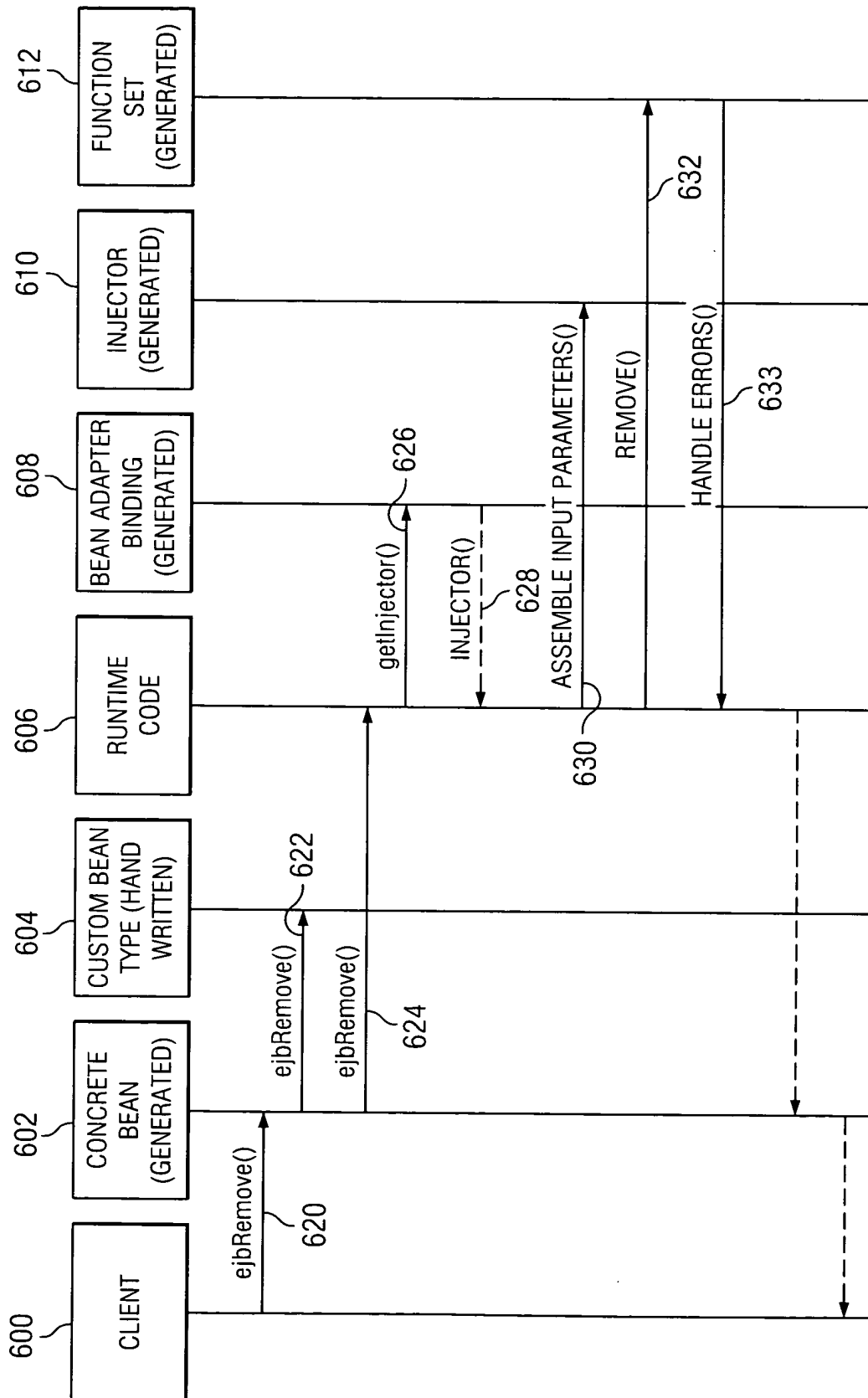
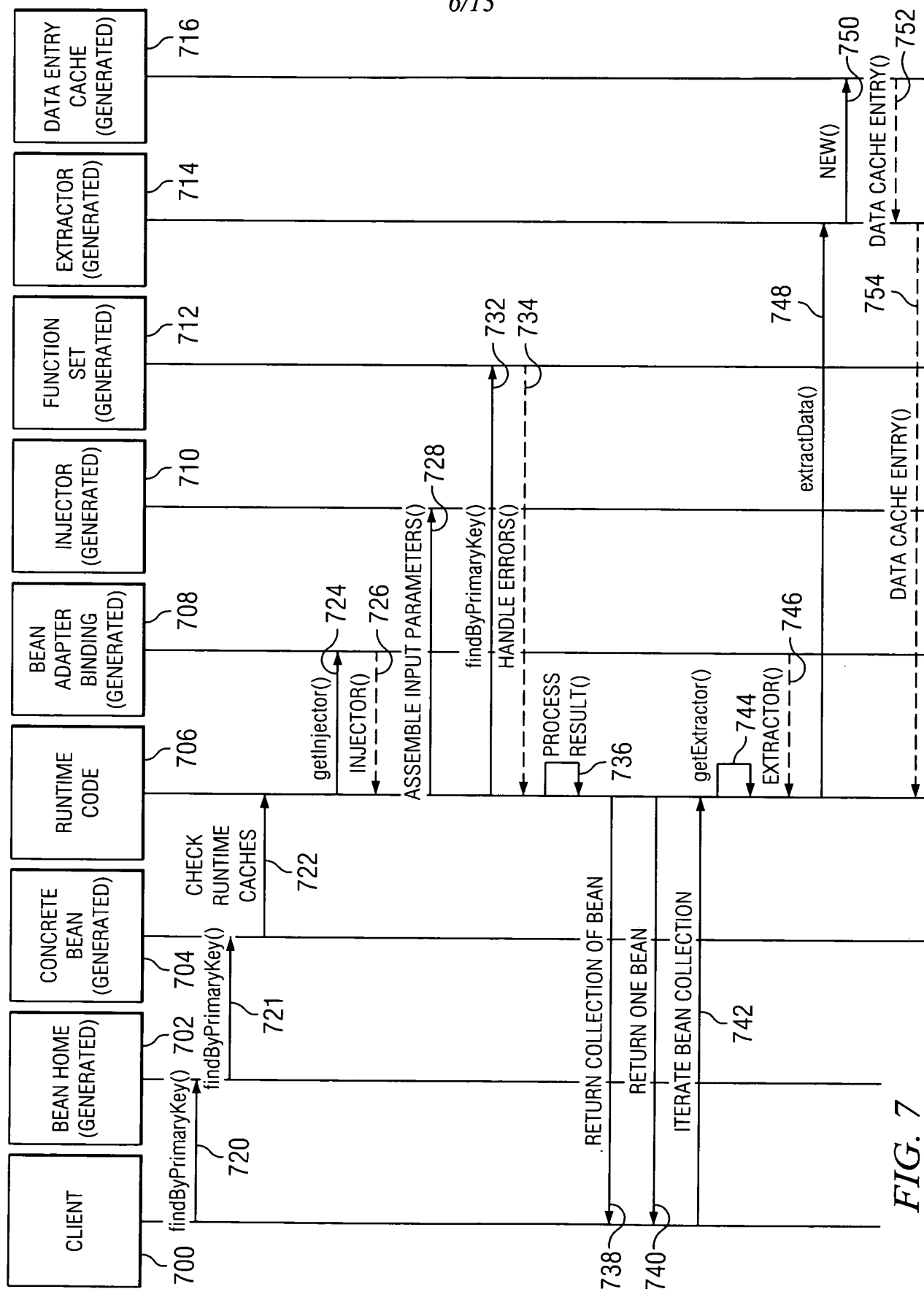


FIG. 6



7/13

```

public class ConcreteCustomerEJB_4fb7b8ee 900
    extends com.ibm.ws.pmttest.model.base.CustomerBean 902
    implements javax.ejb.EntityBean, ConcreteBeanWithMMLink {
/**
 * @generated
 */
    private ConcreteBeanInstanceExtension instanceExtension; 904
/**
    *ConcreteCustomerEJB_4fb7b8cc
    * @generated
    */
    public ConcreteCustomerEJB_4fb7b8ee() {
        super();
        instanceExtension
            ConcreteBeanInstanceExtensionFactory.getInstance(this);
    }
/**
    * ejbRemove
    * @generated
    */
    public void ejbRemove() throws javax.ejb.RemoveException { 906
        super.ejbRemove();
        instanceExtension.ejbRemove();
    }
/**
    * ejbStore
    * @generated
    */
    public void ejbStore() { 908
        super.ejbStore(); 920
        instanceExtension.ejbStore(); 922
    }

/**
    * ejbFindByPrimaryKey
    * @generated
    */
    public java.lang.Integer ejbFindByPrimaryKey(java.lang.Integer arg0) 910
        throws javax.ejb.FinderException {
        return (java.lang.Integer) instanceExtension.ejbFindByPrimaryKey(arg0);
    }
/**
    * ejbCreate
    * @generated
    */
    public java.lang.Integer ejbCreate( 912
        java.lang.Integer arg0,
        java.lang.String arg1)
        throws javax.ejb.CreateException {
        super.ejbCreate(arg0, arg1);
        return (java.lang.Integer) instanceExtension.ejbCreate();
    }
}

```

FIG. 9

8/13

```

public class CustomerAdaptorBinding
    implements com.ibm.ws.ejbpersistence.beanextensions.BeanAdaptorBinding {
    /**
1002  * getExtractor
    * @generated
    */
    public com.ibm.ws.ejbpersistence.dataaccess.EJBExtractor getExtractor() { 1004
        int beanChunkLength = 2;
        // extractor for
com.ibm.ws.pmtest.model.base.websphere_deploy.DB2UDBNT_V72_1.CustomerExtractor 1006
        com.ibm.ws.ejbpersistence.dataaccess.AbstractEJBExtractor extractor =
            new com.ibm.ws.pmtest.model.base.websphere_deploy.DB2UDBNT_V72_1
                .CustomerExtractor();
        extractor.setChunkLength(beanChunkLength);
        extractor.setPrimaryKeyColumns(new int[] { 2 });
        extractor.setDataColumns(new int[] { 1, 2 });
        return extractor;
    }
    /**
    * getInjector
    * @generated
    */
    public com.ibm.ws.ejbpersistence.beanextensions.EJBInjector getInjector() { 1006
        return new com.ibm.ws.pmtest.model.base.websphere_deploy.DB2UDBNT_V72_1
            .CustomerInjectorImpl();
    }
    /**
    * createDataAccessSpecs
    * @generated
    */
    public java.util.Collection createDataAccessSpecs() 1010
        throws javax.resource.ResourceException {
        com.ibm.ws.ejbpersistence.beanextensions.EJBDataAccessSpec daSpec;
        com.ibm.ws.rsadapter.cci.WSInteractionSpecImpl iSpec;
        java.util.Collection result = new java.util.ArrayList(15);

        daSpec =
            com
                .ibm
                .ws
                .ejbpersistence
                .beanextensions
                .DataAccessSpecFactory
                .getDataAccessSpec();
        iSpec = new com.ibm.ws.rsadapter.cci.WSInteractionSpecImpl();
        iSpec.setFunctionSetName(
            "com.ibm.ws.pmtest.model.base.websphere_deploy.DB2UDBNT_V72_1
1012 CustomerFunctionSet");
        iSpec.setFunctionName("Create");
        daSpec.setInteractionSpec(iSpec);
        daSpec.setSpecName("Create");
        daSpec.setInputRecordName("Create");
        daSpec.setOptimistic(false);
        daSpec.setType(
            com
                .ibm
                .ws
                .ejbpersistence
                .beanextensions
                .EJBDataAccessSpec
                .CREATE_BEAN);
        daSpec.setQueryScope(new String[] { "Customer" });
        result.add(daSpec); 1014
    }
}

```

FIG. 10



9/13

```

public class CustomerInjectorImpl 1100
    implements com.ibm.ws.pmttest.model.base.websphere_deploy.CustomerBeanInjector {
    /**
    * ejbCreate
    * @generated
    */
    public void ejbCreate( 1102
        com.ibm.ws.ejbpersistence.beanextensions.ConcreteBean cb,
        javax.resource.cci.IndexedRecord record( {
        com.ibm.ws.pmttest.model.base.ConcreteCustomerEJB_4fb7b8ee concreteBean =
            (com.ibm.ws.pmttest.model.base.ConcreteCustomerEJB_4fb7b8ee) cb;
        record.set(0, concreteBean.getName()); 1110
        record.set(1, concreteBean.getId()); 1112
    }
    /**
    * ejbStore
    * @generated
    */
    public void ejbStore( 1104
        com.ibm.ws.ejbpersistence.beanextensions.ConcreteBean cb,
        javax.resource.cci.IndexedRecord record) {
        com.ibm.ws.pmttest.model.base.ConcreteCustomerEJB_4fb7b8ee concreteBean =
            (com.ibm.ws.pmttest.model.base.ConcreteCustomerEJB_4fb7b8ee) cb;
        record.set(0, concreteBean.getName());
        record.set(1, concreteBean.getId());
    }
    /**
    * ejbRemove
    * @generated
    */
    public void ejbRemove( 1106
        com.ibm.ws.ejbpersistence.beanextension.ConcreteBean.cb,
        javax.resource.cci.IndexedRecord record) {
        com.ibm.ws.pmttest.model.base.ConcreteCustomerEJB_4fb7b8ee concreteBean =
            (com.ibm.ws.pmttest.model.base.ConcreteCustomerEJB_4fb7b8ee) cb;
        record.set(1, concreteBean.getId());
    }
    /**
    * ejbFindByPrimaryKey
    * @generated
    */
    public void ejbFindByPrimaryKey( 1108
        Object pkeyObject,
        javax.resource.cci.IndexedRecord record) {
        java.lang.Integer pkey - (java.lang.Integer) pkeyObject;
        record.set(0, pkey);
    }
}

```

FIG. 11

```

public class CustomerExtractor 1200 10/13
    extends com.ibm.ws.ejbpersistence.dataaccess.AbstractEJBExtractor {
/**
 * CustomerExtractor
 * @generated
 */
    public CustomerExtractor() {
        int[] pkCols = { 2 };
        setPrimaryKeyColumns(pkCols);

        int[] dataCols = { 1,2,3 };
        setDataColumns(dataCols);
    }
/**
 * extractData
 * @generated
 */
    public com.ibm.ws.ejbpersistence.cache.DataCacheEntry extractData( 1202
        com.ibm.ws.ejbpersistence.dataaccess.RawBeanData dataRow)
        throws
            com.ibm.ws.ejbpersistence.utilpm.ErrorProcessingResultCollectionRow,
            com.ibm.ws.ejbpersistence.utilpm.PersistenceManagerInternalError {
        int[] dataColumns = getDataColumns();

        com.ibm.ws.pmttest.model.base.websphere_deploy.DB2UDBNT_V72_1
            .CustomerCacheEntryImpl entry =
                new com
                    .ibm
                    .ws
                    .pmttest
                    .model
                    .base
                    .websphere_deploy
                    .DB2uDBNT_V72_1
                    .CustomerCacheEntryImpl(); 1204

        entry.setDataForNAME(dataRow.getString(dataColumns[0]));
        entry.setDataForID(dataRow.getInt(dataColumns[1]), dataRow.isNull());

        return entry;
    }
/**
 * extractPrimaryKey
 * @generated
 */
    public Object extractPrimaryKey( 1206
        com.ibm.ws.ejbpersistence.dataaccess.RawBeanData dataRow)
        throws
            com.ibm.ws.ejbpersistence.utilpm.ErrorProcessingResultCollectionRow,
            com.ibm.ws.ejbpersistence.utilpm.PersistenceManagerInternalError {
        int[] primaryKeyColumns = getPrimaryKeyColumns();

        java.lang.Integer key;
        key new Integer(dataRow.getInt(primaryKeyColumns[0]));

        return key;
    }
}

```

FIG. 12

```

public class CustomerCacheEntryImpl 1300
    extends com.ibm.ws.ejbpersistence.cache.DataCacheEntry
    implements com.ibm.ws.pmttest.model.base.websphere_deploy.CustomerBeanCacheEntry {
    /**
    * @generated
    */
    private String NAME_Data;
    /**
    * @generated
    */
    private int ID_Data;
    /**
    * @generated
    */
    private boolean ID_IsNull;
    /**
    * getName
    * @generated
    */
    public java.lang.String getName() { 1304
        return NAME_Data;
    }
    /**
    * setDataForNAME
    * @generated
    */
    public void setDataForNAME(String data) { 1308
        this.NAME_Data = data;
    }
    /**
    * getId
    * @generated
    */
    public java.lang.Integer getId() { 1306
        if(ID_IsNull)
            return null;
        else
            return new Integer(ID_Data);
    }
    /**
    * setDataForID
    * @generated
    */
    public void setDataForID(int data, boolean isNull) { 1310
        this.ID_Data = data;
        this.ID_IsNull = isNull;
    }
}

```

*FIG. 13*

```

public class CustomerFunctionSet 1400 12/13
    extends com.ibm.ws.rsadapter.cci.WSResourceAdapterBase
    implements com.ibm.websphere.rsadapter.DataAccessFunctionSet {
/**
 * @generated
 */
    private java.util.HashMap functionHash; 1404
/**
 * Create
 * @generated
 */
    public void Create( 1402
        IndexedRecord inputRecord,
        Object connection,
        WSInteractionSpec interactionSpec)
        throws DataStoreAdapterException {
        PreparedStatement pstmt = null;
        try {
            pstmt =
                prepareStatement(
                    connection,
                    "INSERT INTO CUSTOMEREJB (ID, NAME) VALUES
(?)");

            //For column NAME
            {
                String tempString;

                tempString = (String) inputRecord.get(0);
                if(tempString == null)
                    pstmt.setNull(2, java.sql.Types.VARCHAR);
                else
                    pstmt.setString(2, tempString);
            }

            //For column ID
            {
                Integer tempInteger;

                tempInteger = (Integer) inputRecord.get(1);
                if(tempInteger == null)
                    pstmt.setNull(1, java.sql.Types.INTEGER);
                else
                    pstmt.setInt(1, tempInteger.intValue());
            }

            if (pstmt.executeUpdate() == 0)
                throw new DataStoreAdapterException(
                    "DSA_ERROR",
                    new javax.ejb.NoSuchEntityException(),
                    this.getClass());
        } catch (SQLException e) {
            throw new DataStoreAdapterException(
                "DSA_ERROR",
                e,
                this.getClass());
        } finally {

```

*FIG. 14A*

/\*\*

\* CustomerFunctionSet

\* @generated

\*/

public CustomerFunctionSet() {

1406 functionHash = new java.util.HashMap(15);

```

        functionHash.put("Create", new Integer(0));
        functionHash.put("Remove", new Integer(1));
        functionHash.put("Store", new Integer(2));
        functionHash.put("StoreUsingOCC", new Integer(3));
        functionHash.put("FindByName", new Integer(4));
        functionHash.put("FindByNameForUpdate", new Integer(5));
        functionHash.put("FindByPrimaryKey", new Integer(6));
        functionHash.put("FindByPrimaryKeyForUpdate", new Integer(7));
        functionHash.put("ejbSelectAllNames", new Integer(8));
        functionHash.put("ejbSelectCustomersNamed", new Integer(9));
        functionHash.put("ejbSelectCustomersNamedForUpdate", new Integer(10));
        functionHash.put("findCustomersByAliasesKey_Local", new Integer(11));
        functionHash.put(
            "findCustomersByAliasesKey_LocalForUpdate",
            new Integer(12));
        functionHash.put("CreateCustomersAliasesTuple", new Integer(13));
        functionHash.put("RemoveCustomersAliasesTuple", new Integer(14));

```

1408

}

/\*\*

\* execute

\* @generated

\*/

public Record execute( 1410

WSInteractionSpec interactionSpec,

IndexedRecord inputRecord,

Object connection)

throws javax.resource.ResourceException {

String functionName = interactionSpec.getFunctionName();

Record outputRecord = null;

int functionIndex =

((Integer) functionHash.get(functionName)).intValue();

switch (functionIndex) {

case 0:

Create(inputRecord, connection, interactionSpec);

break;

case 1:

Remove(inputRecord, connection, interactionSpec);

break;

case 2:

Store(inputRecord, connection, interactionSpec);

break;

case 3:

StoreUsingOCC(inputRecord, connection, interactionSpec);

break;

case 4:

outputRecord =

FindByName(inputRecord, connection, interactionSpec);

break;

FIG. 14B